DOCUMENT RESUME

ED 334 223 TM 016 704

AUTHOR	Freeman, Jared; And Others
TITLE	A Knowledge Elicitation Technique for Educational Development: The Critical Decision Method.
PUB DATE	Apr 91
NOTE	12p.; Paper presented at the Annual Meeting of the American Educational Research Association (Chicago, IL, April 3-7, 1991).
PUB TYPE	Reports - Research/Technical (143) -- Speeches/Conference Papers (150)
EDRS PR?CE	MF01/PC01 Plus Postage.
DESCRIPTORS	Critical Incidents Method; Data Collection; Debugging (Computers); *Educational Development; Heuristics; *Interviews; *Knowledge Level; *Measurement Techniques; *Programers; Training Methods
IDENTIFIERS	*Critical Decision Method; Expertise; *Experts; Structured Interviews; Subject Content Knowledge

ABSTRACT

The Critical Decision Method (CDM) is a structured interview method for eliciting expert knowledge. The method was used in a study of computer programmers who were experts at debugging complex computer systems. Fifteen programmers who were identified by their supervisors as experts were asked to describe an experience in which their expertise made a difference. The CDM interviewer used a set of questions to elaborate the programmers' responses. Two non-expert programmers were interviewed for comparison. Each subject told the story of a debugging experience four times with varying degrees of intervention by the interviewers. Data were validated by a panel of eight subjects who reviewed the findings of the interviews. The results show that the CDM facilitated the identification and description of expert skills (critical knowledge) as well as resources and heuristics for selection and use of skills and resources by the experts. The stories produced have the potential to enhance training materials for other programmers. A 32-item list of references is included. (SLD)

A Knowledge Elicitation Technique for Educational Development: The Critical Decision Method

Jared Freeman, Teachers College, Columbia University

Julian Weitzenfeld, J.S. Weitzenfeld Consulting, East Windsor, NJ

Gary Klein, Klein Associates, Yellow Springs, OH

Tom Riedl, Riedl Consulting, New York, NY

John Musa, AT&T Bell Laboratories, Murray Hill, NJ

Paper presented at the annual conference of the
American Educational Research Association
Chicago, Illinois
April 6, 1991

Abstract

The Critical Decision Method (CDM) is a structured interview method for eliciting expert knowledge. The method was employed in a study of computer programmers who are expert at debugging complex computer systems. The role of the CDM in that study is described and the method is critiqued.

Introduction

Developers of artificial intelligence systems commonly use interviewing techniques to study expert performance. Training developers conduct interviews to develop course content material, though this is done too rarely in general, and too rarely involves experts in particular. In artificial intelligence (AI), the resulting products have often proved limited in scope and fragile in character (i.e., they recover poorly, if at all, from trivial errors). Though evaluated less-often and less-publicly, educational products may have similar failings.

These failings are in part due to the ways that researchers use interviews to elicit expert knowledge. LaFrance (1989) has proposed three reasons that interview-based projects often fail. Interviewers may:

- Artificially restrict the breadth of the knowledge domain under investigation.
- Fail to grasp the complexity of experts' goals.
- Fail to identify the wide range of facts and knowledge structures critical to expert performance.

The interviewing methods employed in AI and educational development projects are of two types: unstructured and structured.

An extreme example of the unstructured approach is the autobiographical interview (Langness and Frank, 1981), in which the interviewer simply asks the subject to describe some event or period in detail. Such interviews are unstructured in that the interviewer's choice of follow-up questions is largely ad hoc, lacking theoretical foundation. These methods can elicit rich descriptions of the social and physical environments in which problems are solved and of the skills and knowledge brought to bear upon them.

However, unstructured interviews can incur exorbitant costs for sampling across sites, sampling across subjects, and data analysis. Unstructured methods tend to elicit highly compiled (Anderson, 1985) descriptions of skills, knowledge of whose components may be critical to an audience of students (or AI programmers). Finally, interview probes may vary in each interview, so inte.actions between questions and subjects or questions and subject matter are constant threats to validity.

A less radical variety of unstructured interview is the think-aloud protocol (Ericsson and Simon, 1980). It is a concurrent, or on-line method in which the subject is asked to speak while working on some experimental task, typically in a laboratory. In their review of interview-based research, Ericsson and Simon found that instances of unreliable or invalid data in studies employing think-aloud protocols could be explained by demonstrating that the information requested was not in working memory during the problem-solving episode. Thus, think-aloud protocols concerning subjects' concurrent experiences are assumed to produce data of high validity and reliability. However, the laboratory settings common to think-aloud studies artificially constrain the definition of problems and the resources with which they are solved. (Notably, laboratory tasks often restrict such resources as time, job-aids, and human advice). As Rogoff (1984) has noted, this is unrealistic insofar as real-world problems arise in situ and are constrained by broader, real-world contexts.

Structured interviews resolve some of these problems. The "flexible" interview (Ginsburg, 1987, 1983) is a refinement of the Piagetian clinical interview. Used for diagnosing children's mathematics skills, the method involves presenting a subject with a problem, asking the child to talk while solving it, and then probing for details of method with a limited range of open-ended questions such as:

- "How did you decide 12 plus 9 make 21?"

- "What did you do to get that answer?"

- "How would you teach someone to solve that problem?"

The interviewer then presents problems which test the range and consistency of application of the subject's reported method.

The flexible interview elicits a detailed description of the processes of problem solving. However, it is efficient only in domains where correct answers are known and solution methods are cataloged (such as elementary mathematics).

The method explored in this paper is the Critical Decision Method (CDM), a variant on the Critical Incident Technique (CIT)(Flanagan, 1954). The CDM was created to identify strategies used to make rapid decisions involving high stakes (Klein, Calderwood, and Clinton-Cirocco, 1986) for training and training needs assessment (Klein, Calderwood, and MacGregor, 1988). Subjects are asked to describe an experience in which their expertise made a difference. The CDM interviewer employs a small set of probes, or questions (described below) to elaborate the story. The probes help to identify environmental cues the expert employs in recognitional decision-making (Klein, 1989), the actions the expert takes, the options the expert considers, and the expert's criteria for choosing among those alternatives. The method has been applied to industrial studies of critical infant care, fire-fighting, military action, and consumer-product purchasing decisions.

There is some overlap between the CDM and unstructured methods. For instance, Langness and Frank (1981) note that autobiographical interviews tend to elicit critical incidents.

> "Exceptional experiences...are detailed... And, presumably, these are events that strongly affected the author's sense of self because, as one critic suggests, the author of an autobiography would have no reason to write one unless some sort of inner transformation had occurred." pg. 89.

Langness herself has used a method similar to the CDM, called event analysis (Langness and Frank, 1981) to examine incidents that facilitate or hinder "normal" life style for the mildly retarded. Like the CDM, event analysis examines incidents in detail, but it elicits stories across a lifetime (rather than within recent months or years), and involves interviews with all available participants.

The remainder of this paper will describe the CDM as it was applic d to a study of how expert programmers debug massive computer sy tems. Various authors have noted the failure of psychological research to address the central complexities of computer program debugging. Brooks (1980) observes that code used as experimental material is unrealistically brief. Sheil (1981) states that the laboratory setting biases subjects' use of tools and selection of goals. (In fact it often dictates goal selection and promotes contaminating ancillary goals, such as pleasing the experimenter). Pennington (1982) has asserted that this research contains many contradictory findings at dubious levels of significance. As Pennington has stated,

> "There are no clear, consistent effects for any of the variables [from language features through practices such as using flow charts and on-line tools]. There is more individual difference than experimentally manipulated variability."

The current project attempted to identify some of the complexities that have so far confounded psychological research into program debugging, and it was designed to gain insight into how experts grapple with these complexities. The goal of the project was to produce materials for a training program that could boost the skills of several thousand programmers working for AT&T Bell Laboratories. That goal was met. A course entitled "How Experts Debug Complex Systems" is currently being taught to well-enrolled classes throughout Bell Laboratories.

The Expert Debugging Project and the CDM

Sampling

Representative sampling is a common stumbling block in research involving the time of scarce and costly experts. It was a primary consideration in this project. A survey of staff programming managers was conducted across the corporation. The survey indicated wide variability in the information provided at the outset of debugging assignments, the tools used, and the size of the software products. Accordingly, the CDM interviews were diversified by project site and product size. Interviews were conducted with 15 programmers identified by their supervisors as "experts." The researchers provided three criteri by which supervisors assessed expertise:

- Programmers to whom the supervisors gave complex problems;

- Programmers to whom they referred other programmers; and

- Programmers who debug systems more efficiently.

Two programmers identified by supervisors as non-experts were also interviewed. One expert was interviewed on two occasions (concerning the same cases). The remainder were interviewed once. All interviews lasted three to four hours, in which time each expert told two or three stories. The research team was blind to the expertise level of most of the subjects during the interviews. However, expertise was self-evident. The researchers were able to independently assess the expertise of all subjects with high accuracy in an informal poll.

Interview Methodology

At the beginning of each the CDM interview, the principal interviewer introduced himself and one to three other researchers attending the session. All of the researchers were cognitive psychologists, two had programming experience. They recorded interviews on videotape and took notes on computer. Those with programming experience acted as technical translators for the principal interviewer and methodology expert, when necessary. The interviewer told each subject the purpose of the study, that a supervisor had recommended him or her as a source, and that the content of the interview was confidential.

The interviewer then asked the expert to recall and describe a debugging incident in which his or her skill was particularly important. Each story was told four times.

In the first recitation, the expert briefly told the debugging story, with little or no interruption. This typically took five to ten minutes.

In the second iteration, the interviewer presented the story in the form of a timeline on a whiteboard. The interviewer and subject jointly edited it. The timeline listed environmental stimuli (e.g., "The system erroneously reported lack of memory"), the subject's associated actions (e.g., "I examined the system's memory allocation scheme"), and the timeframe of each action.

In the third instance, the interviewer employed a set of probes to identify decision points, options and decision criteria. The interviewer elaborated the timeline accordingly. The following is a list of the questions used, though the interviewer varied their selection, wording, and sequence of probes:

- What did you know at this time?

- What options did you consider?

- Why did you choose this option?

- What experiences or training were needed?

- Was anyone else involved? What did they do?

- What materials or tools did you use? How did you use them?

In the fourth recitation of the story, experts identified potential pitfalls. This information highlighted critical decision points, options, and decision criteria. A variety of questions facilitated this process, among them several recommended by LaFrance (1988). The interviewer elicited detailed descriptions of problem characteristics by asking naive questions or by positing the question, "What if I had been handed the problem at this stage, unskilled as I am in this field?" Challenging expert reports was occasionally productive. Using this strategy, the interviewer played devil's advocate or sought exceptions to the generalizations experts made. The interviewer also appealed to the experts' experience as mentors with probes such as "How might a less-experienced person have erred here?"

A fifth type of recitation phase is under development. It is designed to produce videotapes of expert stories for classroom use. Tapes of previous recitations tend to be unsatisfactory for several reasons. The first recitation is often incomplete, the second iteration is dominated by an interviewer, the third is too lengthy, and the fourth conveys only partial and speculative accounts. Furthermore, the latter few recitations tend to employ dense terms whose meaning the expert and interviewer have refined throughout the interview. This phenomenon is widely recognized in the social psychology literature (Isaacs, E.A., and Clark, H.H., 1987; Krauss, R.M., 1987; Kraut, R.E., Lewis, S.H., and Swezey, L., 1982). In the proposed fifth iteration, the interviewer and subject discuss the language appropriate to the classroom audience and rehearse the explanation of key incidents if necessary. The interviewer then uses the timeline as a score with which to "conduct" the expert in a final recitation.

Analysis

In the analysis phase, the research team reviewed videotapes, transcripts, timelines, and personal notes to identify expert decision-making patterns and the resources experts employ to debug systems.

Several decision-making strategies were identified, each of which is illustrated below:

- Recognitional decision making -- Experts often made decisions using perceptual and recognitional cues, rather rational, analytical processes. This is typical of experts studied in other fields, such as chess (DeGroot, 1978).

* Backward chaining -- Experts' stories indicate that when the expert knows where a system failed, he or she may search execution paths backward from the failure state to a known "good" state.

* Forward chaining - When the point of failure is a mystery or cannot be reproduced, experts typically reason forward from the system's starting state to all possible states, hoping to find one that produces the failure symptoms.

* Progressive deepening -- Experts conducting forward or backward searches in well-known code sometimes used the intelligent search strategy called progressive deepening (DeGroot, 1978). Using this method, the expert uses judgement or recognitional capacities to select actions on a breadth-first bases, explores them to an arbitrary depth, and continues re:ursively. Debugging experts were observed to use this strategy in backward chaining patterns, as well as the forward-chaining patterns that DeGroot observed.

* Explanation-based decisions -- Experts used their experience to sort through the available data and symptoms to construct a story of how the failure could be occurring. The story identifies the flaw at some level of specificity, and serves as a basis for suggesting tests and debugging procedures. These in turn modify, confirm, or reject the story and provide evidence for a better story. Pennington and Hastie (1986) have presented the explanation-based decision model to account for the behavior of jurors assessing guilt.

* Simulation -- Kahneman and Tversky (1982) have described a process of mental simulaticn to imagine the consequences of course of action. Programmers simulated compilation or machine execution of the compiled code to test how a suspected error could account for a set of symptoms, as well as to test ar outcome of a troubleshooting strategy.

Experts typically used several of these strategies on a single debugging assignment. In one such instance, a programmer used progressive deepening by ranking into three classes (could not, probably would not, or was likely to have prompted an error message) the functions that called a complaining function. He started with the likeliest functions, identified the ancestors of the error-reporting function, and proceeded backwards up through the possible execution paths using backward chaining until he detected the flaw. He also employed forward analysis to test one hypothesis with on-line execution-tracing tools, having previously employed mental simulation to anticipate how this might work.

Analysis of decision making in the CDM data can also address rarer trategies. For example, "elimination by aspects" (Tversky, 1972) enables the expert to eliminate options that possess some characteristic or fail to meet some criterion. "Multiattribute utility analysis" enables the expert to select the option that maximizes some group of attributes.

The analysis phase revealed resources that experts use in their work and some of the key conditions under which they use them. In contrast to the resources provided in programming studies, experts in realistic environments commonly employ documents (including requirements, specifications, manuals, and the system code), run-time debugging tools, and people (including other experts and clients). The researchers focussed on heuristics that experts apply when using specific resources (e.g., when hunting for suspected timing bugs, be aware of the impact on timing of tracing and breakpointing tools themselves) as well as heuristics that appear to operate over all resources (e.g., run easy tests first).

The researchers also considered the character of the bugs experts discussed. Contrary to our expectations, the most salient bugs to experts were not the sort of subtle semantic programming errors (e.g., a stray semicolon or missing default case) catalogued in the few texts on debugging, but larger-scale design flaws (e.g., omission of program logic or poorly sequenced actions). The focus of the course was altered accordingly.

Finally, the analysis phase served as an occasion to identify persuasive and coherent stories that were eventually used in the debugging course.

Flanagan (1954) recommends a highly structured approach to CIT analysis. Its steps are to develop categories of behavior, validate them, select behaviors to categorize, and categorize them. The researchers of the current project followed this method insofar as they employed psychologically validated strategies as categories and intentionally classified debugging behaviors at a high level.

Data Validation

Data validation was conducted by a panel of eight experts chosen from among the subjects. During a day-long session, the experts reviewed and refined general findings derived from the debugging stories described in interviews.

In preparation for the panel session, the team circulated among the experts a summary of the findings, and a questionnaire that listed major debugging strategies (e.g., forward and backward chaining) and techniques (e.g., breakpointing, use of truth tables) mentioned in the interviews or the literature. Experts indicated how often they used each method, how critical it was, and how effective. The responses (when weighted and summed) produced a ranked list of debugging techniques. The panel opened with a round-robin discussion of the highest rated of these methods, then moved on to discussions of broader skills and strategies.

The panel session was productive in several respects, and is recommended as a validation phase in CDM studies of expert behavior.

First and foremost, the panel clarified and approved the team's findings. For example, expert debugging is a less linear process than was implied in most of the individual interviews, in which time pressures or the linear character of the timeline may have biased the data. An illustration is the expert habit of frequently pausing between detailed tests to reexamine a problem at the highest level. In addition, the panel session highlighted the importance of emotional cues that received scant attention in the analysis phase. For example, several experts described the sense of incipient thrashing that signals them to seek out other experts for advice.

Gathering experts from many sites into a single room helped to resolve questions about the use of strategies across environments. For example, the interviews suggested that only a few sites had discovered the debugging heuristic, "Run easy tests first." The panel session revealed that no tests are easy in environments with highly complex systems or where laboratory time is scarce and laboratory tests involve difficult set-up of equipment. The rule is known in such environments, however.

Second, the panel session produced new technical tips and anecdotes, as well as videotaped testimonials concerning the importance of skills critical to expertise (e.g., interviewing techniques), but not valued by non-experts.

A third benefit of the panel session was that the experts' conversations set a standard for the level of technical terminology that the authors could reasonably expect students to grasp. Experts' comments in the panel were generally far less technical -- and thus suited to a broader audience -- than their language in on-site interviews concerning specific system failures.

Finally, the experts endorsed the course developers' strategy of teaching non-expert debuggers high-level strategies, rather than language- or system-specific tricks.

Thus, expert review of CDM findings can refine and validate findings, produce new data, set standards for the language to be used in training materials, and validate strategies for the use of data. It should be noted that reviews should, ideally, be conducted by experts who have not participated in interviews. In this case, however, the interviews had so involved the expert subject that every one agreed to attend the review at the expense of his own department. Given a limited budget and the perceived objectivity of the experts involved, the compromise seemed acceptable.

Benefits of the CDM

Experts' support of the research findings suggests that the Critical Decision Method elicited valid data. There are several reasons one would expect the CDM to do so.

Episodic memory, from which expert stories arise, has proven to be highly reliable and valid (Tulving, 1972; Bower, Black, and Turner, 1979). Evidence that memory is hierarchically structured (Kintsch et al., 1975; Thorndyke, 1977; Chase and Ericsson, 1981) suggests that critical incidents within a story should be recalled accurately. Ericsson and Simon (1980) caution that subjects may not accurately recall the details of a single instance of an often-repeated task. However, the CDM does not elicit routine incidents.

Ericsson and Simon (1980) have reported that verbal protocols should exhibit high validity when intermediate cognitive processing is excluded. The CDM largely controls two of the three intermediate processes cited by Ericsson and Simon; it handles the third moderately well.

- Editing -- Editorial filtering is minimized because each subject is asked to report everything he or she recalls concerning a single event. The timeline provides an additional signal of and a check against editing, as discussed below. Edgerton and Langness (1978) have introduced another check by eliciting descriptions of each event from several participants. However, the salience of the incident and its details may vary by subject. Thus, the reliability of reports may be poor across experts. It is interesting to note that subjects in the debugging study did not seem to edit their stories at the highest level. Specifically, they did not appear biased towards selecting success stories. The interviews elicited a number of incidents in which the expert failed to learn the root cause of the problem, and resolved it by building safeguards against its symptoms.

- Inferencing -- Ericsson and Simon (1980) warn that "Interpretive probing, unlike the critical incident technique, cannot be relied on to produce data stemming directly from the subjects' actual sequences of thought processes." (pg. 221). For example, asking a subject why she took some action produces less reliable data than asking what actions she took. CDM subjects are asked to report what they observed or concluded during a single incident, not their assessments of events nor generalizations across incidents. Thus, inferential and generative errors should be minimized. Subjects in the debugging study used generalization (e.g., "I do this whenever I see symptoms like that.") principally as a last resort, to explain decision-making at the recognitional level.

- Intermediate recoding -- Information that is not stored verbally may not be verbalized accurately, according to Ericsson and Simon (1980). Subjects of the CDM are invited not only to speak, but to draw pictures to describe parts of the task that are best represented visually. By interviewing subjects on the job site, the CDM interviewer allows the subject to use tools and other media to convey concepts that are difficult to verbalize or draw. Videotaping is especially important for analysis of such material because it captures visual, aural, and concurrent events more accurately than can a researcher's written notes.

Does forgetting bias CDM reports? The effect seems to be minimal. Crandall (1989) reports that the CDM proved as evocative as think-aloud protocols in terms of the number of plans and actions, and the specificity of goals elicited. Similarly, Flanagan (1954) asserts, on the basis of numerous CIT studies, that reports of salient historical events are reasonably complete. Ericsson and Simon (1980) declined to address the accuracy of retrieval from long-term memory, though they warn that the accuracy of retrospective reports may suffer from forgetting or compilation that obscures details. While forgetting almost certainly affects CDM reports to some extent, the use of the timeline signals memory or reporting problems to the interviewer. Unoccupied slots on the timeline point the interviewer toward unreported events. Overlapping events in a time slot may indicate inaccurate recall. Both conditions mark the need for intensive questioning. Finally, using the timeline to represent the problem in its totality seems to help subjects recall details.

The CDM has other benefits, in addition to the validity of the reports it elicits.

Like unstructured interviews, the CDM allows subjects to define the domain and the nature of the expertise they exercise over it. For example, the results of this debugging study challenged assumptions common in psychological experiments in this field.

- Each of thirty incidents involved social interaction. In laboratory experiments debuggers almost always work alone.

- Most incidents involved the use of system tools or documentation. Few experimental situations do so. (Gould, 1975, is an exception).

- The experts in the current study had as much as twenty years of professional experience (not counting graduate education). Most studies employ seniors or graduate students from computer science departments as experts.

- The code discussed by debugging experts ranged in size from ten lines (previously selected by colleagues from within a massive system) to hundreds of thousands of lines. Laboratory experiments uniformly have employed code that is 500 lines or shorter (rarely more than 250 lines), and written languages that promote a less dense programming style than C, the one discussed by many of the experts interviewed in the current project.

- The debugging experts used information concerning the run-time and testing hardware as well as the reliability of code authors and users. This information has not been provided subjects in laboratory experiments, to our knowledge.

As expected of a clinical interview, the CDM facilitates decomposition of expert stories into skills, such as how experts rank search paths during backwards analysis.

The CDM may elicit fairly reliable reports. In the instance in which a subject repeated two stories at a second interview session, the reports were consistent.

The CDM elicits eloquent reports, even from professionals whose principle claim is technological knowledge, not interpersonal communication skill. The story-telling method seems to generate enthusiasm, as indicated by the general tenor of the interviews and, more concretely, by the willingness of experts to attend the distant, final panel session at their own expense.

Finally, the CDM is economical. Interviews take as little as two hours each. Videotapes of interviews serve both as data and as course materials.

Disadvantages

The CDM does have several shortcomings.

Flanagan (1954) points out the difficulty of developing a classification scheme for observed behaviors. He advises only that the scheme reflect the application of the findings. Thus, research intended to facilitate personnel selection might categorize behaviors in terms of personality traits. In the current project, targeted towards training, behaviors were categorized by strategy, resources used, and heuristics regarding resource selection and use.

Cross-site differences are difficult to analyze, though judicious sampling can ensure that they are at least evident. As mentioned previously, the rule "run cheap tests first" was not employed in some environments. Only panel session interactions explained why.

Recall of details may not be accurate. Low-level discrepancies (e.g., the number of programming variables) were found between subject recall and program code in the one instance which was investigated in detail. Those discrepancies did not, however, invalidate significant aspects of the expert's recall.

The CDM is difficult to use with people who have a general and theoretical knowledge of a domain, but little direct experience. These people simply don't have stories to tell. This was not the case in the debugging study. "Hands-off" experts can contribute rules and declarative knowledge that, if valid, is best elicited using other methods.

The CDM elicits the critical fringe of challenges – the ones that challenge experts. It does not elicit information about common problems, except insofar as they are components of the rarer incidents. The resea.ch sponsor must determine at the outset whether there is a greater marginal return for facilitating the solution of common problems or the uncommon ones that the CDM elicits.

Finally the CDM does not identify the optimal intervention. Thus, it can provide materials for training, but no direction concerning their comparative value, nor concerning the organization or medium that best represents them.

Summary

The Critical Decision Method is an economical and productive method of structured interview. It is particularly appropriate for exploratory investigations of expertise in complex domains. There is evidence (Crandall, 1989) that CDM interviews produce valid and reliable reports of problem-solving incidents, and there is a solid theoretical foundation for such a claim (Ericsson and Simon, 1980). This study did not test these claims, however. The CDM facilitates decompilation of expert skills. It aids identification of critical knowledge and resources, as well as heuristics for their selection and use. In addition, the method produces stories that can enrich training materials by virtue of their content, their episodic format, and the respect that students attribute to the expert sources.

We hope that other researchers will explore the method. Quantitative tests of the CDM would be particularly useful. However, application of the method to a broader range of domains will almost certainly reward both researchers in those domains and those who wish to refine knowledge elicitation techniques such as the Critical Decision Method.

###

Bibliography

Anderson, Richard C. and Pearson, P. David. (1984). A schema-theoretic view of basic processes in reading comprehension. In Pearson, P.D. (Ed.), Handbook of Reading Research. NY: Longman.

Anderson, John R. (1985). Cognitive Psychology and its Implications. New York: W.H. Freeman and Co.

Bower, Gordon H., J.B. Black, and T.J. Turner. (1979). Scripts in memory for text. Cognitive Psychology, 11, 177-220.

Brooks, Ruven E. (1980). Studying programmer behavior experimentally: The problems of proper methodology. Communications of the ACM, 23(4), 207-213.

Chase, William G. and Ericsson, K. Anders. (1981). Skilled memory. In John R. Anderson, (Ed.), Cognitive Skills and Their Acquisition. Hillsdale, NJ: Erlbaum.

Crandall, Beth W. (1989). A comparative study of think-aloud and critical decision knowledge elicitation methods. SIGART Newsletter, April (108), 144-145.

DeGroot. (1978). Thought & Choice in Chess. The Hague: Mouton (Second edition).

Ericsson, K. A., and Herbert Simon. (1980). Verbal reports as data. Psychological Review, 87(3), 215-251.

Flanagan, John C. (1954). The critical incident technique. Psychological Bulletin, 51(4), 327-358.

Ginsburg, Herbert P. (Ed.), (1983). The Development of Mathematical Thinking. New York: Academic Press, Inc.

Ginsburg, Herbert (1989). Children's Arithmetic: How They Learn It and How You Teach It. (2nd ed.) Austin: Pro-Ed.

Gould, John D. (1975). Some psychological evidence on how people debug computer programs. International Journal of Man-Machine Studies, 7, 151-82.

Isaacs, E.A., and Clark, H.H. (1987). References in conversation between experts and novices. Journal of Experimental Psychology: General, 116, 26-37.

Kahneman, D., and Tversky, A. (1982) The simulation heuristic. In D. Kahneman, P. Slovic, and A Tversky (Eds.), Judgement Under Uncertainty: Heuristics and Biases. Cambridge, MA: Cambridge University Press.

Kintsch, W., Kozminsky, E., Streby, W.J., McKoon, G., and Keenan, J.M. (1975). Comprehension and recall of text as a function of content variables. Journal of Verbal Learning and Verbal Behavior, 14, 196-214.

Klein, G.A., Calderwood, R., and Clinton-Cirocco, A. (1986). Rapid decision making on the fire ground. Proceedings of the Human Factors Society 30th Annual Meeting, 1, 576-580. Dayton, OH: Human Factors Society.

Klein, G.A. (1989). Recognition-primed decisions. In W. Rouse, (Ed.), Advances in Man-Machine Systems Research, 5, 47-92. Greenwich, CT: JAI Press, Inc.

Klein, G.A. (1990). Knowledge Engineering: Beyond expert systems. In M. Singh and A. Sage (Eds.), Information and Decison Technologies, 16(1). North Holland: Elsevier Science Publishers BV.

Klein, Gary A. Peio Karen J. 1989. Use of a prediction paradigm to evaluate proficient decision making. American Journal of Psychology, 102(3), 321-331.

Klein, Gary; Calderwood, Roberta; and MacGregor, Donald. (1989). Critical decision method for eliciting knowledge. IEEE Transactions on Systems, Man, and Cybernetics 19(3), 462-472.

Krauss, R.M. (1987). The role of the listener: Addressee influences on message formulation. Journal of Language and Social Psychology, 6, 81-97.

Kraut, R.E., S.H.L., and Swezey, L. (1982). Listener responsiveness and the coordination of conversation. Journal of Personality and Social Psychology, 43, 718-731.

LaFrance, Marianne. (1988). The knowledge acquisition grid: A method for training knowledge engineers. In Gaines, B.R. and Boose, J.H. Knowledge Acquisition for Knowledge-Based Systems. New York, Academic Press.

LaFrance, Marianne. (1989). The quality of expertise: implications of expert-novice differences for knowledge acquisition. SIGART Newsletter, 108, 6-14.

Langness, L.L. and Frank, Gelya. (1981). Lives: An Anthropological Approach to Biography. Novato, CA: Chandler & Sharp Publishers.

Pennington, N. and Hastie, R. (1986). Evidence evaluation in complex decision making. Journal of Personality and Social Psychology, 51, 242-258.

Rogoff, Barbara. (1984). Thinking and learning in social context. In Barbara Rogoff and Jean Lave (Eds.), Everyday Cognition: Its Development and Social Context. Cambridge, Mass.: Harvard University Press.

Schank, Roger C. (1990). Tell Me a Story: A New Look at Real and Artificial Memory. NY: Charles Scribner's Sons.

Sheil, B.A. The psychological study of programming. ACM Reprint No. 0010-4892/81/0300-0121. 1981.

Thorndyke, P.W. (1977). Cognitive structures in comprehension and memory of narrative discourse. Cognitive Psychology, 9, 77-110.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving and W. Donaldson (Eds.), Organization of Memory. New York: Academic Press.

Tversky, A. (1972). Elimination by aspects: A theory of choice. Psychologyical Review, 79(4), 281-299.